

WHAT IS CLAIMED IS:

1 1. A process comprising:
 2 identifying a first method and a second method to be performed on an object, wherein
 3 the object corresponds to an instantiation of a class;
 4 developing the first method in a first application having a first subclass of the class, wherein
 5 a first application-specific object is an instantiation of the first subclass;
 6 concurrently developing the second method in a second application having a second subclass of
 7 the class, wherein
 8 a second application-specific object is an instantiation of the second subclass.

1 2. The process of claim 1 further comprising:
 2 invoking the first method, wherein
 3 the invoking performs the first method on the first application-specific object such that
 4 the object communicates as if the first method were performed on the object.

1 3. The process of claim 1 further comprising:
 2 invoking the second method, wherein
 3 the invoking performs the second method on the second application-specific object such
 4 that the object communicates as if the second method were performed on the
 5 object.

1 4. The process of claim 1 further comprising:
 2 modifying the first method, wherein the modifying does not affect the second method.

1 5. The process of claim 1 further comprising:
 2 modifying the second method, wherein the modifying does not affect the first method.

1 6. A process comprising:
 2 defining an abstract class for an object, the abstract class comprising:
 3 a first method calling a first application; and
 4 a second method calling a second application;
 5 developing the first method in a first subclass of the abstract class in the first application; and
 6 developing the second method in a second subclass of the abstract class in the second
 7 application.

1 7. An architecture comprising:
2 an object corresponding to an instantiation of a class;
3 a first application having a first subclass of the class, wherein
4 a first application-specific object is an instantiation of the first subclass;
5 the first subclass comprises a first method comprising a first behavior of the first
6 application-specific object; and
7 the first behavior of the first application-specific object corresponds to a first behavior of
8 the object;
9 a second application having a second subclass of the class, wherein
10 a second application-specific object is an instantiation of the second subclass;
11 the second subclass comprises a second method comprising a second behavior of the
12 second application-specific object; and
13 the second behavior of the second application-specific object corresponds to a second
14 behavior of the object.

1 8. The architecture of claim 7 wherein
2 invoking the first method performs the first method on the first application-specific object such
3 that the object communicates as if the first method were performed on the object.

1 9. The architecture of claim 7 wherein
2 invoking the second method performs the first method on the second application-specific object
3 such that the object communicates as if the second method were performed on the object.

1 10. The architecture of claim 7 wherein
2 modifying the first method does not affect the second method.

1 11. The architecture of claim 7 wherein
2 modifying the second method does not affect the first method.

1 12. A computer program product comprising:
2 programming environment instructions for providing a programming environment comprising:
3 identifying instructions to identify a first method and a second method to be performed
4 on an object, wherein
5 the object corresponds to an instantiation of a class;

6 developing instructions to develop the first method in a first application having a first
 7 subclass of the class, wherein
 8 a first application-specific object is an instantiation of the first subclass;
 9 concurrent developing instructions to concurrently develop the second method in a
 10 second application having a second subclass of the class, wherein
 11 a second application-specific object is an instantiation of the second subclass;

12 and

13 a computer-readable medium to store the programming environment instructions, the identifying
 14 instructions, the developing instructions, and the concurrent developing instructions.

1 13. The computer program product of claim 10 wherein
 2 invoking the first method performs the first method on the first application-specific object such
 3 that the object communicates as if the first method were performed on the object.

1 14. The computer program product of claim 10 wherein
 2 invoking the second method performs the first method on the second application-specific object
 3 such that the object communicates as if the second method were performed on the object.

1 15. The computer program product of claim 10 wherein
 2 modifying the first method does not affect the second method.

1 16. The computer program product of claim 10 wherein
 2 modifying the second method does not affect the first method.